

CLAIMS

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A virtualizer, comprising:
 - a task manager for coordinating execution of primary data commands received from a host system;
 - a cache subsystem coupled to said task manager for receiving data requests corresponding to said primary data commands and reconciling said data requests to determine if the cache subsystem can service said data requests; and
 - a command mapper coupled to said cache subsystem for receiving data requests that cannot be serviced by the cache subsystem and parsing said data requests into at least one secondary data command to be sent to a downstream data storage element.
2. The virtualizer according to claim 1, further comprising a target port for receiving said primary data commands from said host system.
3. The virtualizer according to claim 1, further comprising an initiator port for outputting said secondary data commands to said downstream data storage element.
4. The virtualizer according to claim 1, wherein said target port is coupled to said task manager and said cache subsystem.
5. The virtualizer according to claim 3, wherein said virtualizer has multiple ones of said initiator port for interfacing to multiple downstream data storage elements.
6. A method of operating a virtualizer, said method comprising:
 - receiving a primary data command from an external host system;
 - forwarding said primary data command to a task manager;

coordinating, in said task manager, execution of said primary data command;

forwarding a data request corresponding to said primary data command from said task manager to a cache subsystem, said cache subsystem reconciling said data request determine if the cache subsystem can service said data request;

retrieving data from said cache subsystem and forwarding said retrieved data to said host subsystem, if said cache subsystem has said requested data and can service the data request;

forwarding said data request from said task manager to a command mapper, if said cache subsystem does not have said requested data;

parsing said data request in said command mapper into at least one secondary data command; and

forwarding said secondary data command from said command mapper to a downstream data storage element.

7. A method of operating a virtualizer architecture, said method comprising:

issuing a write data command for data on a volume of a data storage element, said data storage element coupled to a first of a plurality of redundant arrays of independent disk (RAID) controllers;

receiving said write data command, and data corresponding to said write data command, in a first virtualizer;

storing said write data in a cache subsystem of said first virtualizer;

copying said write data into a cache of a second virtualizer; and

forwarding the write data with a write command from said second virtualizer to said first RAID controller.

8. A method of operating a virtualizer architecture, said method comprising:

issuing a write data command from a host processor for data on a disk, said disk coupled to a first of a plurality of virtualizers;

receiving said write data command, and write data corresponding to said write data command, in said first virtualizer;

storing said write data in a cache subsystem of said first virtualizer;

copying said write data into a cache of a second virtualizer; and

forwarding the write data and a write command to said disk.

9. The method according to claim 8, wherein said write command comprises a coalesced write to sequential logical block addresses (LBAs) of said disk.

10. A method of operating a virtualizer architecture, said method comprising:

issuing, from a host processor, a primary read data command for data on volumes of multiple data storage elements, each of said data storage elements coupled to one of a plurality of redundant arrays of independent (RAID) controllers;

receiving, in one of a plurality of virtualizers, said primary read data command;

generating multiple secondary read data commands for said data stored on volumes of multiple data storage elements;

forwarding said multiple secondary read data commands to said volumes of said multiple data storage elements;

receiving, in said RAID controllers, said requested data from said volumes of multiple data storage elements via a storage interconnect;

forwarding, from said RAID controllers, said requested data to said one virtualizer; and

storing said requested data in a cache subsystem of said one virtualizer; and

forwarding, from said cache subsystem of said one virtualizer, said requested data to said host processor.

11. A virtualizer architecture, comprising:

at least one host processor;

a plurality of virtualizers for processing primary data commands from said at least one host processor, said plurality of virtualizers coupled to said at least one host processor via a host interconnect;

a plurality of redundantly paired redundant arrays of independent disk (RAID) controllers coupled to said plurality of virtualizers via an interconnect fabric and a virtualization layer interconnect, each of said plurality of virtualizers including a cache memory subsystem for temporarily storing data associated with data access requests; and

at least one data storage element coupled to one of said RAID controllers of each of said redundantly paired RAID controllers, each data storage element coupled to said one of said RAID controllers via a storage interconnect, each one of said pair of redundantly paired RAID controllers coupled to each other via an interconnect.

12. The virtualizer architecture according to claim 11, wherein said plurality of virtualizers have read-caching and read-ahead functionality.